UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/620,747	07/16/2003	Mark S. Moir	6000-33600	8970

58467 7590 01/08/2009

MHKKG/SUN
P.O. BOX 398
AUSTIN, TX 78767

EXAMINER	
TECKLU, ISAAC TUKU	
ART UNIT	PAPER NUMBER
2192	
MAIL DATE	DELIVERY MODE
01/08/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/620,747	**Applicant(s)** MOIR ET AL.	
	Examiner ISAAC T. TECKLU	**Art Unit** 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 October 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-43 and 45-59* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-43 and 45-59* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 44 and 45 have been cancelled.

2. Claims 1-43 and 46-59 have been reexamined.

Continued Examination Under 37 CFR 1.114

3. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 10/20/2008 has been entered.

Response to Arguments

4. Applicant's arguments filed 04/10/2008 have been fully considered but they are not persuasive.

The Applicant asserted that "Daynes does not disclose *a method of providing non-blocking multi-target transactions in a computer system*" (Page 13).

The applicant asserted that "Daynes describes a non-blocking synchronization for changing the lock state associated with a single one of these resources (see, e.g., column 17, lines

2 – column 18, line 13). However, this clearly does not provide a non-blocking multi-target transactions, since this mechanism is used to access a single target, the lock state associated with a single shared resource. Even if a non-blocking synchronization primitive were used to change the state of multiple locks, this would not teach the non-blocking multi-target transactions of Applicants' claim, as the transaction itself (which use locks to mage access to resources) is clearly a blocking operation." (pages 12 and 13)

Examiner respectfully disagrees with the above assertion. Contrary to Applicant's assertion, prior art of record, Daynes discloses a method of providing non-blocking multi-target transactions in a computer system (e.g. FIG. 13, col.15:64-67 and col.16:1-10 "… locking operation may make use of two techniques….dispatching specialized code according to lock state type, and non-blocking synchronizations, emphasis added). Daynes discloses a method for locking by sharing lock states. Each resource or object has an association lock state comprised of transactions that own a lock in a specific mode for the resource. The objective of Daynes invention, inter alia is to provide lock free mechanism to avoid a situation called deadlock. For convenience of Applicant, examiner cites certain related portion of the prior art below:

Col.2:15-25 describes: "One type of lock is a shared lock. A shared lock permits multiple transactions to read (view) an item simultaneously without any modification or addition to the item (no writing is permitted). A shared lock is referred to as permitting concurrent (or concurrency) control by a transaction (i.e., multiple transactions are permitted to concurrently access a resource). Another type of lock is an exclusive lock. An exclusive lock permits one transaction to read and write to an item while excluding all other transactions from reading or writing to the item" (emphasis added).

Col.2:25-40 describes: "The locking and unlocking of resources must be administered to ensure that any required lock properties are complied with. For example, two or more different transactions cannot each acquire an exclusive lock at the same time for the same resource. Additionally, locks must be administered to provide a queue for transactions that are waiting to acquire a lock, and to rollback any executed actions if a deadlock results (i.e., when each of two transactions are waiting for a lock release from the other before continuing). For example, a deadlock occurs if transaction 1 has a lock on resource A and is waiting to acquire a lock on resource B, and transaction 2 has a lock on resource B and is waiting to acquire a lock on resource A." (emphasis added).

Col.2:40-45 describes: "A locking protocol partially determines the administration of a locking and unlocking of resources. A locking protocol determines how and when a transaction is granted (or acquires) a lock for a resource and when the resource is unlocked (i.e., the lock is released allowing other transactions to acquire a lock on that resource). A lock manager administers a locking protocol" (emphasis added).

Col.13:15-30 describes; "Referring to step 800 of FIG. 8, the value of the new lock state that the resource will be associated with is computed. The new value is obtained by removing the transaction from any owner set of the lock state where it appears in. Referring to the pseudo code, to obtain the new value, the current value of the lock state is obtained and copied into a local variable (lines 3-4). The new value is obtained by removing the transaction from the owner sets (of the lock states indicated in the local variable) (lines 5-9 of the pseudo code). At step 802 and line 10 of the pseudo code, the new lock state is obtained from the TILS. At step 804 and line 11, the resource's association with the TILS is updated to reflect the retrieved lock state. Transactions waiting in the queue for the lock release are then processed at step 806 and line 15 of the pseudocode. These waiting transactions will resume at step 708 of FIG. 7" (emphasis added).

It is clear that Dynes clearly provide a non-blocking multi-target transaction (from the above cited portions and col.17:25-40 "... Non-blocking synchronizations... non-blocking synchronization requires an implementation using an atomic compare and swap operation). During garbage collection, the garbage collector checks if any owner sets of the lock state (to be copied to another location) contains inactive bit numbers (e.g., bits that mapped to locking contexts of terminated transactions (as described above) that did not delete their bits from the lock states representing the locks these transactions owned upon their completion). Thus examiner concludes that Daynes clearly teaches non-blocking multi-target transactions to avoid deadlocking/convoying condition.

The Applicant asserted that " Daynes does not disclose *defining plural tranasactionable locations, wherein individual ones of the transactionable locations encode respective values* ***and are owned by no more than one transaction at any given point in a unthreaded computation."*** (Page 14).

Regarding to the above assertion, examiner respectfully disagrees with the above assertion. Contrary to the above argument, Daynes clearly discloses defining plural transactionable locations, wherein individual ones of the transactionable locations encode respective values and are owned by no more than one transaction at any given point in a unthreaded computation (col.2:15-25, see above). Even though Daynes describes one type of lock to be "shared lock which permits multiple transactions to read/view an item simultaneously without any modification", Daynes also describes another type of lock referred to as "exclusive lock which permits one transaction to read and write to an item while excluding all other transactions from reading or writing to the item" (col.2:15-25). Thus, it is respectfully submitted that the above argument is not persuasive.

The Applicant asserted that "Daynes also fails to disclose *wherein the ownership acquiring wrests ownership from another transaction, if any, that owns the targeted transactionable location."* (pages 14 and 15).

Examiner respectfully disagrees. Daynes describes ownership acquiring wrests ownership from another transaction, if any, that owns the targeted transactionable location (e.g. FIG. 9 and col.13:40-60 "… for each lock state found, the transaction is removed from all owner sets where it appears"). By removing the transaction from the owner sets, the value of the lock state is modified."). By removing the transaction from the owner, the ownership acquiring wrests ownership from another transaction. Thus, it is respectfully submitted that the above argument is not persuasive.

The Applicant asserted that "Daynes fails to disclose *attempting to commit the particular multi-target transactions using a single-target transaction continues to own each of the targeted transactionable locations."* (Page 16).

Regarding to the above assertion, the examiner respectfully disagrees. The claim limitation recites "attempting …" interpreted as "testing, trying etc". Daynes describes attempting to commit the particular multi-target transactions using a single-target transaction continues to own each of the targeted transactionable locations (e.g. FIG. 7). "If the transaction does not own the requested lock, the lock manager checks for any conflicts with the existing locks and for the absence of any pending lock requests (by other transactions) at step 702. If there is a conflict but no pending lock request (determined at step 704), a new lock state with a queue is created and entered in the TILS at step 704.a" (col.11:10-30, emphasis added). Furthermore, Daynes describes that "if the compare-and swap fails (given by a test at line 9), it means that at least one other transaction has managed to set its own lock while the transaction was executing instructions at lines 3 to 8. The lock manager must then retry the lock acquisition with the new lock state (lines 13 and 14 initiate the retry and jump to line 4 to redispatch to specialized code best suited to handle the type of the new lock state returned by the compare-and swap instructions). Otherwise, if the compare-and swap succeeds, the lock manager completes the lock acquisition by recording the locked resource in its lock set" (col.17:35-65, emphasis added). Thus, it is respectfully submitted that the above argument is not persuasive. Accordingly, examiner respectfully maintains the previous rejection.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

6. Claims 1-15, 17-32, 34-43, 46-49 and 51-59 rejected under 35 U.S.C. 102(a) as being anticipated by Daynes (US 6,182,186 B2).

Per claim 1 (Currently amended), Daynes discloses a method of providing non-blocking multi-target transactions in a computer system (e.g. FIG. 13, col.15:64-67 and col.16:1-10 "… locking operation may make use of two techniques….dispatching specialized code according to lock state type, and non-blocking synchronizations, emphasis added and e.g. FIG. 7 and related text), the method comprising:

defining plural transactionable locations, wherein individual ones of the transactionable locations encode respective values and are owned by no more than one transaction at any given point in a multithreaded computation (col.2:15-25 and col. 18: 15-25 "… transaction owns an exclusive type of lock …" and e.g. FIG. 4 and FIG. 11, 1148 and related text);

for a particular multi-target transaction of the multithreaded computation, attempting to acquire ownership of each of the transactionable locations targeted thereby (col. 17: 65-67 and col. 18:1-10 "… if the ownership test fails …"), wherein the ownership acquiring wrests ownership from another transaction, if any, that owns the targeted transactionable location <u>without the other transaction releasing ownership</u> (col. 18:45-50 "… single-owner lock states of each locking context…" and e.g. FIG. 9 and col.13:40-60 "… for each lock state found, the transaction is removed from all owner sets where it appears); and

once ownership of each of the targeted transactionable locations has been acquired, attempting to commit the particular multi-target transaction using a single-target synchronization primitive to ensure that, at the commit (col. 19:1-15 "… a locking context is active … bit numbers that appear in its owner …"), the particular multi-target transaction continues to own each of the targeted transactionable locations, wherein individual ones of the multi-target transactions do not contribute to progress of another (col. 20:5-15 "… swo are single owner lock states …single owner lock states corresponding to the transaction this cache belongs to… ").

Per claim 2, Daynes discloses the method of claim 1, wherein the ownership wresting employs a single-target synchronization primitive to change status of the wrested from transaction to be incompatible with a commit thereof (col. 12:30-40 "… lock on the same resource are incompatible with each other …").

Per claim 3, Daynes discloses the method of claim 2, wherein, as a result of the status change, the wrested from transaction fails and retries (col. 17: 65-67 and col. 18:1-10 "… if the ownership test fails …").

Per claim 4, Daynes discloses the method of claim 2, wherein the wrested from transaction is itself a multi-target transaction (col. 2:15-25 "… lock permits multiple transactions to read …").

Per claim 5, Daynes discloses the method of claim 1, further comprising: on failure of the commit attempt, reacquiring ownership of each targeted transactionable location and retrying (col. 17: 65-67 and col. 18:1-10 "… if the ownership test fails …").

Per claim 6, Daynes discloses the method of claim 1, wherein no transaction may prevent another from wresting therefrom ownership of transactionable locations targeted by the active transaction (col. 20:5-15 "… swo are single owner lock states …single owner lock states corresponding to the transaction this cache belongs to… ").

Per claim 7, Daynes discloses the method of claim 1, wherein the ownership acquiring employs a single-target synchronization primitive to update the ownership of the targeted transactionable location (col. 3:10-25 "… lock data structure is updated …" and e.g. FIG. 7, step 714 and related text).

Per claim 8, Daynes discloses the method of claim 1, wherein each encoding of a transactionable location is atomically updateable using a single-target synchronization primitive (col. 10:55-65 "… maintain atomicity …").

Per claim 9, Daynes discloses the method of claim 1, wherein the individual transactionable location encodings further include an identification of the owning transaction's corresponding value for the transactionable location (e.g. FIG. 7, step 708 and related text).

Per claim 10, Daynes discloses the method of claim 1, further comprising: accessing values corresponding to individual ones of the transactionable locations using a wait-free load operation (e.g. FIG. 7, step 714 and related text).

Per claim 11, Daynes discloses the method of claim 1, wherein the transactionable locations directly encode the respective values (e.g. FIG. 7, step 708 and related text).

Per claim 12, Daynes discloses the method of claim 1, wherein the transactionable locations are indirectly referenced (e.g. FIG. 9, step 902 and related text).

Per claim 13, Daynes discloses the method of claim 1, wherein the transactionable locations are encoded in storage managed using a nonblocking memory management technique (e.g. GIG. 13 shows non-blocking synchronization …").

Per claim 14, Daynes discloses the method of claim 1, wherein the transactionable locations, if unowned, directly encode the respective values and otherwise encode a reference to the owning transaction (e.g. FIG. 7, step 708 and related text).

Per claim 15, Daynes discloses the method of claim 1, wherein the single-target synchronization primitive employs a Compare-And-Swap (CAS) operation (col. 17:35-55 "… compare-and swap …").

Per claim 17, Daynes discloses the method of claim 1, wherein the single-target of the single-target synchronization primitive includes at least a value and a transaction identifier encoded integrally therewith (col. 16: 55-67 "… information can include register values …").

Per claim 18, Daynes discloses the method of claim 1, wherein the multi-target transaction has semantics of a multi-target compare and swap (NCAS) operation (col. 17:35-55 "… compare-and swap …").

Per claim 19, Daynes discloses the method of claim 1, embodied in operation of an application programming interface (API) that includes a load operation and an multi-target compare and swap (NCAS) operation (col. 17:35-55 "… compare-and swap …").

Per claim 20, Daynes discloses the method of claim 19, wherein the load operation is wait-free (e.g. FIG. 7, step 714 and related text).

Per claim 21, Daynes discloses the method of claim 1, embodied in operation of an application programming interface (API) that provides transactional memory (e.g. FIG. 1 and related text).

Per claim 22, Daynes discloses a computer-readable storage medium program instructions computer-executable to implement:

a plurality of non-blocking, multi-target transactions (e.g. FIG. 13, col.15:64-67 and col.16:1-10 "… locking operation may make use of two techniques….dispatching specialized code according to lock state type, and non-blocking synchronizations);

wherein the program instructions comprise:

instances of one or more single-target synchronization primitives executable to acquire, for a particular multi-target transaction, ownership of targeted transactionable locations (col.17:35-65 and e.g. FIG. 9 and col.13:40-60 "… for each lock state found, the transaction is removed from all owner sets where it appears) and

a particular single-target synchronization primitive executable_to ensure that, at commit, the particular multi-target transaction continues to own each of the targeted transactionable locations, wherein individual ones of the multi-target transactions do not contribute to progress of another (col. 17: 65-67 and col. 18:1-10 "… if the ownership test fails …")

Per claim 23, Daynes discloses the storage medium claim 22, wherein the program instructions are further executable to implement a concurrent computation, and wherein execution of the concurrent computation invokes the multi-target transactions (col. 2:15-25 "… lock permits multiple transactions to read …")

Per claim 24, Daynes discloses the storage medium of claim 22, wherein to acquire ownership, when performed by a first one of the multi target transactions, wrests ownership from respective other ones of the multi-target transactions, if any, that own respective ones of the targeted transactionable locations (col. 19:1-15 "… a locking context is active … bit numbers that appear in its owner …").

Per claim 25, Daynes discloses the storage medium_of claim 24, wherein to wrest ownership, the program instructions are further executable to implement an instance of a single-target synchronization primitive to change status of a wrested-from transaction to be incompatible with a commit thereof (col. 12:30-40 "… lock on the same resource are incompatible with each other …").

Per claim 26, Daynes discloses the storage medium of claim 25, wherein, as a result of the status change, the program instructions are further executable to complement the wrested-from transaction eventually fails and retries (col. 17: 65-67 and col. 18:1-10 "… if the ownership test fails …").

Per claim 27, Daynes discloses the storage medium of claim 22, wherein no transaction may prevent another from wresting therefrom ownership of transactionable locations targeted by the active transaction (e.g. FIG. 7, step 708 and related text).

Per claim 28, Daynes discloses the storage medium of claim 22, wherein the transactionable locations directly encode the respective values (e.g. FIG. 7, step 708 and related text)..

Per claim 29, Daynes discloses the storage medium_of claim 22, wherein the transactionable locations are indirectly referenced (e.g. FIG. 7, step 708 and related text)..

Per claim 30, Daynes discloses the storage medium of claim 22, wherein the transactionable locations are encoded in storage managed using a nonblocking memory management technique (e.g. FIG. 7, step 714 and related text).

Per claim 31, Daynes discloses the storage medium of claim 22, wherein the transactionable locations, if unowned, directly encode the respective values and otherwise encode a reference to the owning transaction (e.g. FIG. 7, step 708 and related text).

Per claim 32, Daynes discloses the storage medium of claim 22, wherein at least some instances of one or more the single-target synchronization primitive employ a Compare-And-Swap (CAS) operation (col. 17:35-55 "… compare-and swap …").

Per claim 34, Daynes discloses the storage medium of claim 22, wherein at least some of the multi-target transaction have semantics of a multi-target compare and swap (NCAS) operation (col. 17:35-55 "… compare-and swap …").

Per claim 35, Daynes discloses the storage medium of claim 22, embodied as software that includes a functional encoding of operations concurrently executable by one or more processors to operate on state of the transactionable locations (e.g. FIG. 7, step 708 and related text).

Per claim 36, Daynes discloses the storage medium of claim 22, wherein at least some of the multi-target transactions are defined by an application programming interface (API) that includes a load operation and a multi-target compare and swap (NCAS) operation (col. 17:35-55 "… compare-and swap …").

Per claim 37, Daynes discloses the storage medium of claim 22, wherein at least some of the multi-target transactions are defined by an application programming interface (API) that provides transactional memory (e.g. FIG. 1 and related text).

Per claim 38, Daynes discloses storage medium of claim 22, wherein the multi-target transactions are obstruction-free, though not wait-free or lock-free (e.g. FIG. 7, step 714 and related text).

Per claim 39, Daynes discloses the storage medium of claim 22, wherein the implementation does not itself guarantee that at least one interfering concurrently executed multi-target transactions makes progress (e.g. FIG. 7, step 708 and related text).

Per claim 40, Daynes discloses the storage medium of claim 22, wherein the program instructions are further executable to implement a contention management facility configured to facilitate progress in a concurrent computation (e.g. FIG. 4 and related text).

Per claim 41, Daynes discloses the storage medium of claim 40, wherein operation of the contention management facility ensures progress of the concurrent computation (e.g. FIG. 7 and related text).

Per claim 42, Daynes discloses the storage medium of claim 40, wherein the contention management facility is modular such that alternative contention management strategies may be employed without affecting correctness of the implementation (e.g. FIG. 7, step 716 and related text).

Per claim 43, Daynes discloses the storage medium of claim 40, wherein the contention management facility allows changes in contention management strategy during a course of the concurrent computation (e.g. FIG. 7, step 708 and related text).

Per claim 46, Daynes discloses a computer readable storage medium storing program instructions computer-executable to implement (e.g. FIG. 13, col.15:64-67 and col.16:1-10 "… locking operation may make use of two techniques….dispatching specialized code according to lock state type, and non-blocking synchronizations, emphasis added and e.g. FIG. 7 and related text):

instantiation of one or more transactionable locations in shared memory configured to individually encapsulate values that may be targeted by concurrent executions of non-blocking multi-target transactions (col.2:15-25 and col. 18: 15-25 "… transaction owns an exclusive type of lock …" and col.15:64-67 and col.16:1-10 "… locking operation may make use of two techniques….dispatching specialized code according to lock state type, and non-blocking synchronizations, emphasis added and e.g. FIG. 7 e.g. FIG. 4 and FIG. 11, 1148, and. FIG. 13 and related text); and

one or more instances of a non-blocking multi-target transaction that upon execution of a particular instance thereof, attempts to acquire ownership of each of a plurality of transactionable locations targeted thereby and (col. 17: 65-67 and col. 18:1-10 "… if the ownership test fails …" and e.g. FIG. 7 and related text), once ownership of each of the targeted transactionable locations has been acquired, attempts to commit the particular instance using a single-target synchronization primitive to ensure that, at the commit (col. 17: 65-67 and col. 18:1-10 "… if the ownership test fails …"), the particular instance continues to own each of the targeted transactionable locations (col. 18:45-50 "… single-owner lock states of each locking context…" and e.g. FIG. 9 and col.13:40-60 "… for each lock state found, the transaction is removed from all owner sets where it appears…"), wherein execution of no one of the multi-target transaction instances contributes to progress of another (col. 19:1-15 "… a locking context is active … bit numbers that appear in its owner …").

Per claim 47, this is the computer readable medium version of the claimed method discussed above (Claim 7), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Daynes.

Per claim 48, this is the computer readable medium version of the claimed method discussed above (Claim 4), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Daynes.

Per claim 49, this is the computer readable medium version of the claimed method discussed above (Claim 15), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Daynes.

Per claim 51, this is the computer readable medium version of the claimed method discussed above (Claim 17), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Daynes.

Per claim 52, this is the computer readable medium version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Daynes.

Per claim 53, this is the computer readable medium version of the claimed method discussed above (Claim 18), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Daynes.

Per claim 54, this is the computer readable medium version of the claimed method discussed above (Claim 5), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Daynes.

Per claim 55, Daynes discloses a encoding of claim 46, wherein the computer readable medium includes at least one medium selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium (e.g. FIG. 1 and related text).

Per claim 56, this is the apparatus version of the claimed method discussed above (Claim 1), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Daynes.

Per claim 57, this is the computer readable medium version of the claimed method discussed above (Claim 7), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Daynes.

Per claim 58, this is the computer readable medium version of the claimed method discussed above (Claim 2), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Daynes.

Per claim 59, this is the computer readable medium version of the claimed method discussed above (Claim 4), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Daynes.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 16, 33 and 50 rejected under 35 U.S.C. 103(a) as being unpatentable over Daynes in view of Maged et al (“Non-Blocking Algorithms and Preemption-Safe Locking on Multiprogrammed Shared Memory Multiprocessors”, March 1997).

Per claims 16, 33 and 50 Daynes do not explicitly disclose wherein the single-target synchronization primitive employs Load-Linked (LL) and Store-Conditional (SC) operation pair. However, Maged et al. discloses non blocking algorithms (See Section 3, page 5). Figure 2, page 7 shows a non-blocking counter implementation using load-linked/store –conditional. In addition, Maged et al. discloses emulation using load-linked and store – conditional instruction (page 13). Therefore it would have been obvious to employ Load-linked and Store-Conditional operation to read, modify and write a shared location as once suggested by Maged et al. (page 3).

Conclusion

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ISAAC T. TECKLU whose telephone number is (571) 272-7957. The examiner can normally be reached on M-TH 9:300A - 8:00P.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Isaac T Tecklu/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192